

STDP-Compatible Approximation of Backpropagation in an Energy-Based Model

Yoshua Bengio

yoshua.umontreal@gmail.com

*Montreal Institute for Learning Algorithms, University of Montreal,
Montreal H3T 1J4, Quebec, Canada, and Canadian Institute for Advanced Research*

Thomas Mesnard

thomas.mesnard@gmail.com

Computer Science Department, École Normale Supérieure, Paris 75005, France

Asja Fischer

asja.fischer@gmail.com

Institute of Computer Science, University of Bonn, Bonn 53117, Germany

Saizheng Zhang

saizhenglisa@gmail.com

*Montreal Institute for Learning Algorithms, University of Montreal,
Montreal H3T 1J4, Quebec, Canada*

Yuhuai Wu

ywu@cs.toronto.edu

University of Toronto, Toronto, Ontario, M5S 1A1, Canada

We show that Langevin Markov chain Monte Carlo inference in an energy-based model with latent variables has the property that the early steps of inference, starting from a stationary point, correspond to propagating error gradients into internal layers, similar to backpropagation. The backpropagated error is with respect to output units that have received an outside driving force pushing them away from the stationary point. Backpropagated error gradients correspond to temporal derivatives with respect to the activation of hidden units. These lead to a weight update proportional to the product of the presynaptic firing rate and the temporal rate of change of the postsynaptic firing rate. Simulations and a theoretical argument suggest that this rate-based update rule is consistent with those associated with spike-timing-dependent plasticity. The ideas presented in this article could be an element of a theory for explaining how brains perform credit assignment in deep hierarchies as efficiently as backpropagation does, with neural computation corresponding to both approximate inference in continuous-valued latent variables and error backpropagation, at the same time.

1 Introduction

It has been hypothesized numerous times (Hinton & Sejnowski, 1986; Friston & Stephan, 2007; Berkes, Orban, Lengyel, & Fiser, 2011) that, given a state of sensory information (current and past inputs), neurons are collectively performing inference, that is, moving toward configurations that better “explain” the observed sensory data and so minimize the corresponding “surprise” and maximize a form of coherence between sensory observations and internal representations. We can think of the configuration of internal neurons (hidden units or latent variables) as an “explanation” for the observed sensory data.

Under this hypothesis, when an unexpected signal arrives at visible neurons, the other neurons will change their state (from one that is near a stochastic equilibrium) so as to better reflect that change in input.¹ If this error signal is slowly driving visible neurons toward the externally observed value, this creates a perturbation for the whole network. In this article, we consider what happens early in this process, when the perturbation due to the incorrect prediction of visible neurons is propagated toward inner areas of the brain (hidden units and hidden layers). As one of the two main contributions of this work, we show how the propagation of this perturbation is mathematically equivalent to the propagation of activation gradients by the backpropagation algorithm in a deep neural network.

This result assumes that latent variables are continuous valued (unlike in the vanilla Boltzmann machine), making the system an energy-based model similar to the continuous Hopfield net (Hopfield, 1984), defined by an energy function over both continuous visible and hidden units. It also assumes that the neurons are leaky integrators with injected noise. As a consequence, neural computation corresponds to gradually going down the energy while adding noise, which corresponds to running a Langevin Markov chain Monte Carlo (MCMC) as inference mechanism.

By applying the chain rule, the backpropagated activation gradients can be used to estimate the error gradient with respect to the weights of the incoming connections. This leads to a weight update performing gradient descent on the output prediction error which is similar to a temporal form of contrastive Hebbian learning. The update is proportional to the product of the presynaptic firing rate and the temporal rate of change of the postsynaptic firing rate. This brings us to the second main contribution of this work. Biological experiments (Markram & Sakmann, 1995; Gerstner, Kempter, van Hemmen, & Wagner, 1996) suggest that synaptic change in several types of neurons and stimulation contexts depends on the relative timing of presynaptic and postsynaptic spikes. Learning rules that mimic

¹Visible neurons are in contact with the outside world, that is, sensory neurons, or, alternatively, internal neurons that directly reflect the sensory signal.

this observation are called spike-timing dependent plasticity (STDP) rules. We show that our proposed weight update that depends only on firing rates and their temporal derivatives is consistent with the STDP observations, according to a series of simulations. This result provides an interesting machine learning interpretation for STDP, since artificial neural networks are generally cast in terms of firing rates.

This work is only a stepping stone. Several points still need to be elucidated before a complete theory of learning, inference, and credit assignment is elaborated that is both biologically plausible and makes sense from a machine learning point of view for whole networks (with global optimization of the whole network and not being limited to learning of visible neurons that receive a target). See section 6 for a discussion.

We start by explaining the connection between the leaky integrator-based neural computation and Langevin MCMC in energy-based models in section 2. Based on this, we show in section 3 that, starting from a fixed point, early inference in such an energy-based model performs backpropagation of prediction error gradients. In section 4, we analyze the resulting weight update and show that it mimics STDP. Section 5 discusses related work and open questions, and directions of future research are given in section 6. We conclude in section 7.

2 Neural Computation Does Inference: Going Down the Energy

We consider the hypothesis that a central interpretation of neural computation (what neurons do, on a short timescale at which weights can be considered fixed) is that it is performing iterative inference. Iterative inference means that the hidden units \mathbf{h} of the network are gradually changed toward configurations that are more probable given the sensory input \mathbf{v} and according to the current model of the world associated with the parameters of the model. In other words, they are approximately moving toward configurations more probable under some probability distribution $P(\mathbf{h}|\mathbf{v})$ associated with the model and eventually sampling from $P(\mathbf{h}|\mathbf{v})$.

Before making the connection between Boltzmann machines or energy-based models and backpropagation, let us see in more mathematical detail how neural computation could be interpreted as inference.

2.1 Leaky Integrator Neurons. For this purpose, consider the classical leaky integrator neural equation. Let \mathbf{s}_t represent the state of the system at time t , a vector with one element per unit (e.g., representing a single neuron), where $s_{t,i}$ is a real-valued quantity associated with the i th unit, corresponding to a time-integrated voltage potential. Let us denote \mathbf{v}_t for the state of the visible units (i.e., a subset of the elements of \mathbf{s}_t representing the externally driven input) and \mathbf{h}_t for the state of the hidden units (i.e., the remaining elements of \mathbf{s}_t such that $\mathbf{s}_t = (\mathbf{v}_t, \mathbf{h}_t)$). Furthermore, let f be the function that computes a new complete state \mathbf{s}_{t+1} given its previous state \mathbf{s}_t ,

with $f = (f_v, f_h)$ denoting the parts of f that, respectively, output the new states of the visible and hidden units.

Here, we assume that the input units are clamped to an external signal and that the time evolution of the unclamped hidden units follows a leaky integrators equation,

$$\mathbf{h}_{t+1} = f_h(\mathbf{s}_t, \boldsymbol{\eta}_t) = \mathbf{h}_t + \epsilon(R_h(\tilde{\mathbf{s}}_t) - \mathbf{h}_t), \quad (2.1)$$

where $R(\mathbf{s}) = (R_v(\mathbf{s}), R_h(\mathbf{s}))$ represents the network-generated pressure on neurons (i.e., $R_i(\mathbf{s})$ is what the rest of the network asks neuron i to move toward) and the corrupted state $\tilde{\mathbf{s}}_t$ results from synaptic noise and spiking effects. Here, we roughly model this corruption by simple additive noise,

$$\tilde{\mathbf{s}}_t = \mathbf{s}_t + \boldsymbol{\eta}_t, \quad (2.2)$$

and we see that equation 2.1 corresponds to the discretization of a differential equation,

$$\tau \dot{\mathbf{h}} = R_h(\mathbf{s} + \boldsymbol{\eta}) - \mathbf{h}, \quad (2.3)$$

which brings \mathbf{h} exponentially fast (with time constant τ) toward the target value $R_h(\mathbf{s})$, along with the random walk movements brought by the noise $\boldsymbol{\eta}$. We assume that $R_i(\mathbf{s})$ is a weighted sum of input signals coming from the connected neurons into neuron i (although the derivation below does not depend on the specific form of R , only on the assumption that it corresponds to an energy gradient). Note that the standard biological neuron models assume that $R_i(\mathbf{s})$ is a weighted sum of the input spikes (or of the input firing rates).

2.2 Machine Learning Interpretation: Leaky Integration as Langevin MCMC. Why can equation 2.1 be seen as performing inference, moving toward states more probable under $P(\mathbf{h}|\mathbf{v})$? Note, first, that $R_h(\tilde{\mathbf{s}}_t)$ in equation 2.1 represents a guess for a new configuration, with $R_h(\tilde{\mathbf{s}}_t) - \mathbf{h}_t$ being a noisy direction of movement and $R_h(\mathbf{s}_t) - \mathbf{h}_t$ is a noise-free direction.

Consider the following direction of change for the neural state \mathbf{s} ,

$$R(\mathbf{s}) - \mathbf{s} \propto \frac{\partial \log P(\mathbf{s})}{\partial \mathbf{s}} = -\frac{\partial E(\mathbf{s})}{\partial \mathbf{s}}, \quad (2.4)$$

for some energy-based probability distribution $P(\mathbf{s}) \propto e^{-E(\mathbf{s})}$ associated with the model, with energy function $E(\mathbf{s})$. This is an assumption that can easily be fulfilled, for example, if $R(\mathbf{s})$ is the linear combination of some input rates $\rho(\mathbf{s})$, as we will show below. Interestingly, equation 2.4 also emerges in recent work on unsupervised learning using denoising autoencoders and denoising score matching, where $R(\mathbf{s}) - \mathbf{s}$ corresponds to the reconstruction

error (Vincent, 2011; Alain & Bengio, 2013). In that context, $R(\mathbf{s})$ corresponds to the reconstruction function of the autoencoder, and $P(\mathbf{s})$ is the probability density function that it estimates, with the associated energy function $E(\mathbf{s})$.

With this interpretation, the leaky integration neural computation of equation 2.1 seems to follow a Langevin MCMC (Andrieu, de Freitas, Doucet, & Jordan, 2003), since

$$\begin{aligned} \mathbf{s}_{t+1} &= \mathbf{s}_t + \epsilon(R(\tilde{\mathbf{s}}_t) - \mathbf{s}_t) \\ &= \mathbf{s}_t + \epsilon(R(\tilde{\mathbf{s}}_t) - \tilde{\mathbf{s}}_t + \tilde{\mathbf{s}}_t - \mathbf{s}_t) \\ &= \mathbf{s}_t + \epsilon\left(-\frac{\partial E(\tilde{\mathbf{s}}_t)}{\partial \tilde{\mathbf{s}}_t} + \boldsymbol{\eta}_t\right), \end{aligned} \quad (2.5)$$

where for the last line we used equations 2.2 and 2.4 and we see that we are going down the energy gradient of $\tilde{\mathbf{s}}_t$. Hence, from the point of view of the noisy states $\tilde{\mathbf{s}}$, we see that the update equation corresponds to

$$\begin{aligned} \tilde{\mathbf{s}}_{t+1} - \boldsymbol{\eta}_{t+1} &= \tilde{\mathbf{s}}_t - \boldsymbol{\eta}_t - \epsilon \frac{\partial E(\tilde{\mathbf{s}}_t)}{\partial \tilde{\mathbf{s}}_t} + \epsilon \boldsymbol{\eta}_t \\ \tilde{\mathbf{s}}_{t+1} &= \tilde{\mathbf{s}}_t - \epsilon \frac{\partial E(\tilde{\mathbf{s}}_t)}{\partial \tilde{\mathbf{s}}_t} + \boldsymbol{\eta}_{t+1} - (1 - \epsilon)\boldsymbol{\eta}_t, \end{aligned} \quad (2.6)$$

which we recognize as going down the gradient of the energy with learning rate ϵ and adding noise $\boldsymbol{\eta}_{t+1} - (1 - \epsilon)\boldsymbol{\eta}_t$.

2.3 A Possible Energy Function. To fix ideas and illustrate the possibility of a driving function R corresponding to the gradient of an energy function, we propose the following energy function, closely related to the Hopfield net (but with latent variables) or with the Boltzmann machine energy function (but with continuous nonlinearities inserted):

$$E(\mathbf{s}) = \sum_i \frac{s_i^2}{2} - \frac{1}{2} \sum_{i \neq j} W_{i,j} \rho(s_i) \rho(s_j) - \sum_i b_i \rho(s_i), \quad (2.7)$$

where $W_{i,j}$ is the weight between unit i and unit j and ρ is the neural nonlinearity, some kind of monotonic bounded function that outputs a value between 0 and 1 corresponding to a firing rate, as, for example,

$$\rho(s_i) = \begin{cases} 0 & \text{if } s_i < \beta_1 \\ s_i - \beta_1 & \text{if } \beta_1 \leq s_i \leq \beta_2 \\ 1 & \text{if } s_i > \beta_2, \end{cases} \quad (2.8)$$

with thresholds β_1 and β_2 such that $\beta_2 - \beta_1 = 1$ and $\beta_1 < 0 < \beta_2$.

Note that from a biological perspective, the nonlinear transformation ρ can be seen as modeling the monotonically increasing dependency of the actual firing rate (or, equivalently, the probability of firing) on the integrated activity (the expected membrane potential, averaged over the random effects of both pre- and postsynaptic spikes).

With this energy function, the driving function R would be

$$R(\mathbf{s}) = \mathbf{s} - \frac{\partial E(\mathbf{s})}{\partial \mathbf{s}}, \quad \text{with}$$

$$R_i(\mathbf{s}) = \rho'(s_i) \left(b_i + \sum_j W_{i,j} \rho(s_j) \right). \quad (2.9)$$

To obtain this, we have assumed that $W_{i,j} = W_{j,i}$. Otherwise, we would get that

$$R_i(\mathbf{s}) = \rho'(s_i) \left(b_i + \sum_j \frac{1}{2} (W_{i,j} + W_{j,i}) \rho(s_j) \right), \quad (2.10)$$

which automatically symmetrizes the weight matrix when following a weight update like the one we propose below.

This formula for the driving input R is similar to the usual weighted sum of firing rates, except for the novel factor $\rho'(s_i)$, which is 1 if $\beta_1 \leq s_i \leq \beta_2$ and 0 otherwise. This would suggest that when neuron i is saturated (being either shut off or firing at the maximal rate), the external inputs have no impact on its state. The only term that remains in the neural update equation, equation 2.1, is the one that drives the state toward 0, bringing it out of the saturation region and back into a regime where the neuron is sensitive to the outside feedback, so long as $\rho(0)$ is not a saturated value (which is guaranteed by the condition that $\beta_1 \leq 0 \leq \beta_2$). This idea is developed further below.

2.3.1 Fixed-Point Behavior. In particular, it is interesting to note what happens around a fixed point of the state dynamics. Inspection of equation 2.5 reveals that for a fixed point $\widehat{\mathbf{s}}$, it must be the case that

$$\frac{\partial E(\widehat{\mathbf{s}})}{\partial \widehat{\mathbf{s}}} = 0 \Rightarrow R(\widehat{\mathbf{s}}) = \widehat{\mathbf{s}} \Rightarrow f(\widehat{\mathbf{s}}) = \widehat{\mathbf{s}}, \quad (2.11)$$

which means that

$$\widehat{s}_i = \rho'(\widehat{s}_i) \left(b_i + \sum_j W_{i,j} \rho(\widehat{s}_j) \right). \quad (2.12)$$

Let us consider the hypothesis that unit i is saturated: $\rho'(s_i) = 0$. In this case, $R_i(s) = 0$, and the neural update becomes

$$s_{t+1,i} = (1 - \epsilon)s_{t,i},$$

which converges toward $s_{t,i} = 0$ with $t \rightarrow \infty$. Thus, s could not have been a fixed point. Therefore, assuming that $\rho(0)$ is not a saturated value (i.e., that the origin corresponds to a region where the derivative is significantly nonzero, $|\rho'(0)| > 0$), we get that if $\rho'(s_i) = 0$, the network cannot be at a fixed point. Equivalently, if the whole network has converged to a fixed point, all neurons must be in a sensitive regime, which makes them respond to their inputs.

3 Link to Backpropagation

We are now ready to present the first important result of this article: a link between neural computation as inference in an energy-based model and backpropagation of prediction error gradients.

3.1 Propagation of Perturbations. Consider what happens when a network such as we have described is in an equilibrium, that is, when the state is at a fixed point. At that point, according to equation 2.11, the average gradient over the noise of the energy is 0 and weight updates are also 0 on average.

To make the link to supervised backpropagation simpler, let us consider two kinds of visible units: input units x and output units y , such that $v = (x, y)$ and $s = (x, y, h)$. Suppose that we start by letting the network settle to a fixed point \hat{s} with x clamped to the observed input values. Then we obtain an output \hat{y} at the fixed point \hat{s} , where $R_y(\hat{s}) = \hat{y}$, $R_h(\hat{s}) = \hat{h}$. Equivalently, we have that

$$\frac{\partial E(\hat{s})}{\partial \hat{y}} = 0 \quad \text{and} \quad \frac{\partial E(\hat{s})}{\partial \hat{h}} = 0. \quad (3.1)$$

That is, the free units (hidden and output) have settled to a value that is in agreement with the clamped input units.

Now suppose that a target value y is observed and gradually drives the output units from their fixed-point value \hat{y} toward y . This happens because the output units are also leaky integrator neurons, meaning that their state gradually changes based on the input they receive, in the direction of the driving signal. Let us denote

$$\Delta y = \epsilon(y - \hat{y}), \quad (3.2)$$

that initial change of $\hat{\mathbf{y}}$ in the direction of the driving signal \mathbf{y} when going from time step 0 to time step 1 (following the neural update equation, equation 2.1).

Let us consider as a training objective the squared prediction error between $\hat{\mathbf{y}}$ and \mathbf{y} . That would push the global state $\hat{\mathbf{s}}$ away from the equilibrium where it was sitting and into a region where $\frac{\partial E(\hat{\mathbf{s}})}{\partial \hat{\mathbf{s}}}$ is nonzero. It is given by

$$C = \frac{1}{2} \|\hat{\mathbf{y}} - \mathbf{y}\|^2. \quad (3.3)$$

Since, at equilibrium, $R_{\mathbf{y}}(\hat{\mathbf{s}}) = \hat{\mathbf{y}}$, this corresponds to the mismatch between the prediction $R_{\mathbf{y}}(\hat{\mathbf{s}})$ and the target value \mathbf{y} driving the output units, that is,

$$C = \frac{1}{2} \|R_{\mathbf{y}}(\hat{\mathbf{s}}) - \mathbf{y}\|^2 = \frac{1}{2\epsilon^2} \|\Delta \mathbf{y}\|^2. \quad (3.4)$$

Note that

$$\Delta \mathbf{y} = -\epsilon \frac{\partial C}{\partial \hat{\mathbf{y}}}, \quad (3.5)$$

which shows that updating $\hat{\mathbf{y}}$ by $\Delta \mathbf{y}$ exactly corresponds to performing a step of gradient descent on $\hat{\mathbf{y}}$ with respect to the objective C , where ϵ can be seen as a learning rate. Of course, our real interest is in changing the weights, especially the weights of lower layers.

So how would the rest of the network react to this external perturbation? Would hidden neurons also move in the direction of reducing the output prediction error? Let us first consider the top hidden layer \mathbf{h}_l , which consists of units directly connected to the output units \mathbf{y} . Starting from the fixed point, we find that the change $\Delta \mathbf{y}$ with respect to $\hat{\mathbf{y}}$ would propagate to $\hat{\mathbf{h}}_l$ via the update of the neural state, which, in the noise-free condition, is given by

$$\Delta \mathbf{h}_l = \epsilon (R_{\mathbf{h}_l}(\mathbf{x}, \hat{\mathbf{y}} + \Delta \mathbf{y}, \hat{\mathbf{h}}) - \hat{\mathbf{h}}_l). \quad (3.6)$$

With $\Delta \mathbf{y}$ small (arising out of our assumption that the visible units only gradually move toward their target), we can approximate the above by taking the Taylor expansion of $R_{\mathbf{h}_l}$ at the fixed point $(\mathbf{x}, \hat{\mathbf{y}}, \hat{\mathbf{h}})$, which is given by

$$R_{\mathbf{h}_l}(\mathbf{x}, \hat{\mathbf{y}} + \Delta \mathbf{y}, \hat{\mathbf{h}}) = \hat{\mathbf{h}}_l + \frac{\partial R_{\mathbf{h}_l}(\hat{\mathbf{s}})}{\partial \hat{\mathbf{y}}} \Delta \mathbf{y} + o(\epsilon), \quad (3.7)$$

exploiting $R_{h_l}(\widehat{\mathbf{s}}) = \widehat{h}_l$ at the fixed point. Insertion into equation 3.6 yields

$$\Delta h_l = \epsilon \frac{\partial R_{h_l}(\widehat{\mathbf{s}})}{\partial \widehat{\mathbf{y}}} \Delta \mathbf{y} + o(\epsilon^2). \quad (3.8)$$

Hence, using equation 3.5, we have that

$$\Delta h_l = -\epsilon^2 \frac{\partial R_{h_l}(\widehat{\mathbf{s}})}{\partial \widehat{\mathbf{y}}} \frac{\partial C}{\partial \widehat{\mathbf{y}}} + o(\epsilon^2). \quad (3.9)$$

Note that with the assumed layer-wise neural network structure (i.e., having no connections between neurons of the same layer) $o(\epsilon)$ (and $o(\epsilon^2)$, respectively) is zero if the neural nonlinearity ρ fulfills $\rho''(s_i) = 0$. This is the case for the nonlinearity given in equation 2.8, since it is piecewise linear (and thus all the higher derivatives of ρ are zero). Therefore, the o -terms are omitted in the following.

In order to obtain backpropagation, what we would like to get is

$$\Delta h_l \propto \frac{\partial C}{\partial \widehat{h}_l} = \frac{\partial R_y(\widehat{\mathbf{s}})}{\partial \widehat{h}_l} \frac{\partial C}{\partial \widehat{\mathbf{y}}}, \quad (3.10)$$

as resulting from $R_y(\widehat{\mathbf{s}}) = \widehat{\mathbf{y}}$ and the application of the chain rule. This would exactly fit equation 3.9, if we can show that $\frac{\partial R_{h_l}(\widehat{\mathbf{s}})}{\partial \widehat{\mathbf{y}}} = \frac{\partial R_y(\widehat{\mathbf{s}})}{\partial \widehat{h}_l}$, which is done in the following. Note that R is a first derivative of a function related to the energy function, that is,

$$R(\mathbf{s}) = \mathbf{s} - \frac{\partial E(\mathbf{s})}{\partial \mathbf{s}} = \frac{\partial L(\mathbf{s})}{\partial \mathbf{s}}, \quad (3.11)$$

where

$$L(\mathbf{s}) = \frac{1}{2} \|\mathbf{s}\|^2 - E(\mathbf{s}). \quad (3.12)$$

Thus, we get $R_y(\widehat{\mathbf{s}}) = \frac{\partial L(\widehat{\mathbf{s}})}{\partial \widehat{\mathbf{y}}}$, $R_{h_l}(\widehat{\mathbf{s}}) = \frac{\partial L(\widehat{\mathbf{s}})}{\partial \widehat{h}_l}$, and it becomes clear that $\frac{\partial R_{h_l}(\widehat{\mathbf{s}})}{\partial \widehat{\mathbf{y}}}$ and $\frac{\partial R_y(\widehat{\mathbf{s}})}{\partial \widehat{h}_l}$ are cross-derivatives of L . As we know that cross derivatives are symmetric, we have

$$\frac{\partial R_{h_l}(\widehat{\mathbf{s}})}{\partial \widehat{\mathbf{y}}} = \frac{\partial^2 L}{\partial \widehat{\mathbf{y}} \partial \widehat{h}_l} = \left(\frac{\partial^2 L}{\partial \widehat{h}_l \partial \widehat{\mathbf{y}}} \right)^T = \frac{\partial R_y(\widehat{\mathbf{s}})}{\partial \widehat{h}_l}^T. \quad (3.13)$$

We are now ready to rewrite $\Delta \mathbf{h}_l$, equation 3.9, in a form that equates it with a backpropagated gradient:

$$\Delta \mathbf{h}_l = -\epsilon^2 \frac{\partial R_{\mathbf{y}}(\widehat{\mathbf{s}})^T}{\partial \widehat{\mathbf{h}}_l} \frac{\partial C}{\partial \widehat{\mathbf{y}}} = -\epsilon^2 \frac{\partial \widehat{\mathbf{y}}^T}{\partial \widehat{\mathbf{h}}_l} \frac{\partial C}{\partial \widehat{\mathbf{y}}} = -\epsilon^2 \frac{\partial C}{\partial \widehat{\mathbf{h}}_l}.$$

Similarly, the perturbation $\Delta \mathbf{h}_l$ will be transmitted at the next time step to the units $\widehat{\mathbf{h}}_{l-1}$ that are directly connected to $\widehat{\mathbf{h}}_l$ (but not to $\widehat{\mathbf{y}}$). We have

$$\Delta \mathbf{h}_{l-1} = \epsilon \left(R_{h_{l-1}}((x, \widehat{\mathbf{y}} + \Delta \mathbf{y}, \widehat{\mathbf{h}}_1 + \Delta \mathbf{h}_1, \widehat{\mathbf{h}}_{/l})) - \widehat{\mathbf{h}}_{l-1} \right), \quad (3.14)$$

where we use $\widehat{\mathbf{h}}_{/l}$ to denote all hidden layers except the first one. Here, the Taylor expansion yields

$$\begin{aligned} R_{h_{l-1}}((x, \widehat{\mathbf{y}} + \Delta \mathbf{y}, \widehat{\mathbf{h}}_1 + \Delta \mathbf{h}_1, \widehat{\mathbf{h}}_{/l})) &= \widehat{\mathbf{h}}_{l-1} + \frac{\partial R_{h_{l-1}}(\widehat{\mathbf{s}})}{\partial \widehat{\mathbf{y}}} \Delta \mathbf{y} \\ &+ \frac{\partial R_{h_{l-1}}(\widehat{\mathbf{s}})}{\partial \widehat{\mathbf{h}}_1} \Delta \mathbf{h}_1 + o(\epsilon), \end{aligned} \quad (3.15)$$

where the second term on the right-hand side is equal to zero assuming the proposed network structure, since with no connections between \mathbf{h}_{l-1} and \mathbf{y} , we obtain $\frac{\partial R_{h_{l-1}}(\widehat{\mathbf{s}})}{\partial \widehat{\mathbf{y}}} \Delta \mathbf{y} = \frac{\partial E(\widehat{\mathbf{s}})}{\partial \widehat{\mathbf{h}}_{l-1} \partial \widehat{\mathbf{h}}_y} = 0$. Following the same line of arguments as before, we get

$$\Delta \mathbf{h}_{l-1} = \epsilon \frac{\partial \widehat{\mathbf{h}}_l^T}{\partial \widehat{\mathbf{h}}_{l-1}} \Delta \mathbf{h}_l = -\epsilon^3 \frac{\partial \widehat{\mathbf{h}}_l^T}{\partial \widehat{\mathbf{h}}_{l-1}} \frac{\partial C}{\partial \widehat{\mathbf{h}}_l} = -\epsilon^3 \frac{\partial C}{\partial \widehat{\mathbf{h}}_{l-1}}.$$

Analogously, the update of layer $l - k$ in the k th time step is

$$\Delta \mathbf{h}_{l-k} = -\epsilon^{k+2} \frac{\partial C}{\partial \widehat{\mathbf{h}}_{l-k}}. \quad (3.16)$$

Thus, we have shown so far that neural computation as inference in an energy-based model performs backpropagation of (weighted) prediction error gradients.

3.2 Stochastic Gradient Descent Weight Update. The above result is consistent with and inspired by the idea previously proposed by Hinton (2007) that temporal change can encode backpropagated gradients. What

would it take for the $\Delta \hat{h}_k$ at layer k to turn into a stochastic gradient descent (SGD) weight update with respect to the prediction error $C = \|\mathbf{y} - \hat{\mathbf{y}}\|^2$?

Let us assume that $W_{i,j}$ is the weight between neuron i in layer $l - k$ and neuron j in layer $l - k - 1$. Let us further recall that Δh_i , as described by equation 3.16, is proportional to $\frac{\partial C}{\partial h_i}$. If we regard \hat{h}_i as a function of $W_{i,j}$ given by inserting equation 2.9 into equation 2.1 and applying the chain rule, we would get the following weight update:

$$\Delta W_{i,j} = \Delta \hat{h}_i \frac{\partial \hat{h}_i}{\partial W_{i,j}} = -\epsilon^{k+2} \frac{\partial C}{\partial \hat{h}_i} \frac{\partial \hat{h}_i}{\partial W_{i,j}}. \quad (3.17)$$

Since h_i is proportional to $W_{i,j} \rho(s_j)$, we would obtain the following weight update:

$$\Delta W_{i,j} \propto \dot{s}_i \rho(s_j). \quad (3.18)$$

Interestingly, it turns out that such a learning rule allows simulating the relationship between spike timing and synaptic change according to STDP, as shown by a theoretical argument and simulations in the following section. This analysis is valid only in the early stages of the perturbation of propagation. (See Scellier & Bengio, 2016, for a mathematical framework allowing one to correctly extend the gradient computation to the case where the perturbation on the output unit fully propagates to the lower layers until the latter reach a fixed point, and requiring propagating not just the initial perturbation but also reaching a modified fixed point or equilibrium but yielding the same learning rule.)

With the above STDP-compatible learning rule, the change in weights due to the initial perturbation would be approximately proportional to the backpropagation update. However, note that the multiplicative factor ϵ^{k+2} for units h_k at layer k makes the initial changes much slower for the lower layers. This is because the leaky integrator neurons have not had time to integrate the information yet, so practically, it will take on the order of the time constant τ (from equation 2.3) times k for the change in \hat{h}_k to become significant unless we adjust the per layer learning rates accordingly. Interestingly, as Scellier and Bengio (2016) show, the ϵ^{k+2} factor disappears if we go beyond the initial perturbation propagation and allow the network to settle after nudging the output units by $\Delta \mathbf{y}$. This allows one to show that the resulting weight update actually corresponds to stochastic gradient descent on the prediction error, provided the weights are also constrained to remain symmetric: $W_{i,j} = W_{j,i}$.

4 Link to Spike-Timing-Dependent Plasticity

We now present the second main contribution of this work: we analyze the weight update rule given by equations 3.18 in more detail and establish a connection to STDP. With the opposite sign, such a weight update rule was proposed by Hinton and McClelland (1988) well before the STDP behavior was reported. A similar rule, also studied by Xie and Seung (2000), also drew a link to STDP. They started by assuming a particular pattern relating spike timing and weight change and then showed that the resulting weight change could be approximated by the product of the presynaptic firing rate and the temporal rate of change of the postsynaptic firing rate. Instead, we go in the other direction, showing that if the weight change is proportional to the product of the presynaptic firing rate (or simply the presence of a spike) and the temporal rate of change of the postsynaptic activity, then we recover a relationship between spike timing and weight change that has the precise characteristics of the one observed experimentally by biologists. We present both an easy-to-understand theoretical justification for this result (see section 4.1), as well as simulations that confirm it (see section 4.2).

4.1 Spike-Timing-Dependent Plasticity. Spike-timing-dependent plasticity is a central subject of research in synaptic plasticity. However, much more research is needed to solidify the links between STDP and a machine learning interpretation of it at the scale of a whole network, that is, at the scale of a network with “hidden layers” which need to receive a useful training signal. (See Markram, Gerstner, & Sjöström, 2012, for a recent review of the neuroscience research on STDP.)

We present here a simple view of STDP that has been observed at least in some common neuron types and preparations: there is a weight change if there is a presynaptic spike in the temporal vicinity of a postsynaptic spike. This change is positive if the postsynaptic spike happens just after the presynaptic spike (and larger if the timing difference is small) and negative if it happens just before (and, again, larger if the timing difference is small, on the order of a few milliseconds). This is illustrated in Figure 1 from Bi and Poo (2001) based on biological data. The amount of change decays to zero as the temporal difference between the two spikes increases beyond a few tens of milliseconds. We are thus interested in this temporal window around a presynaptic spike during which a postsynaptic spike, before or after the presynaptic spike, induces a change of the weight.

Keep in mind that the above pattern of spike-timing dependence is only one aspect of synaptic plasticity. (See Feldman, 2012, for an overview of aspects of synaptic plasticity that go beyond spike timing, including firing rate, as one would expect from this work, but also synaptic cooperativity—nearby synapses on the same dendritic subtree—and depolarization—due to multiple consecutive pairings or spatial integration across nearby

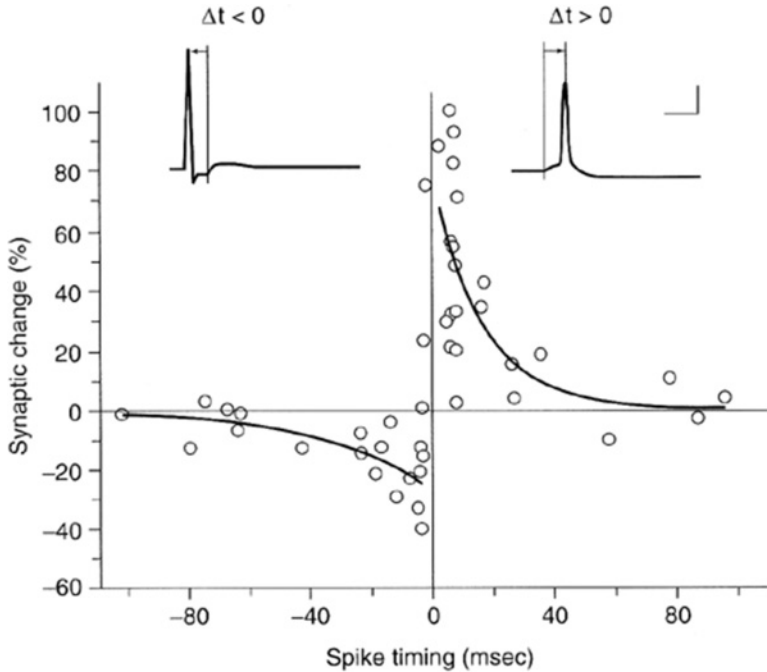


Figure 1: Biological observation of STDP weight change, vertical axis, for different spike timing differences (post minus pre), horizontal axis. From Shepherd (2003), with data from Bi and Poo (2001). Compare with the result of our simulations shown in Figure 4.

locations on the dendrite, as well as the effect of the synapse's distance to the soma.)

4.1.1 Rate-Based Aspect of STDP. A hypothesis inspired by Xie and Seung (2000) and Hinton (2007) is that the STDP weight change can be associated with the temporal rate of change of postsynaptic activity, as a proxy for its association with post minus presynaptic spike times. Both of the above contributions focus on this rate aspect of neural activity, and this is also the case of the synaptic update rule, equation 3.18, studied here.

Since stochastic gradient cares only about the average change instead of equation 3.18, we could also write

$$\Delta W_{i,j} \propto \dot{s}_i \xi_j, \quad (4.1)$$

where ξ_j is the binary indicator of a spike from neuron j , which occurs with an average rate of $\rho(s_j)$. This works, for example, if we assume that the input

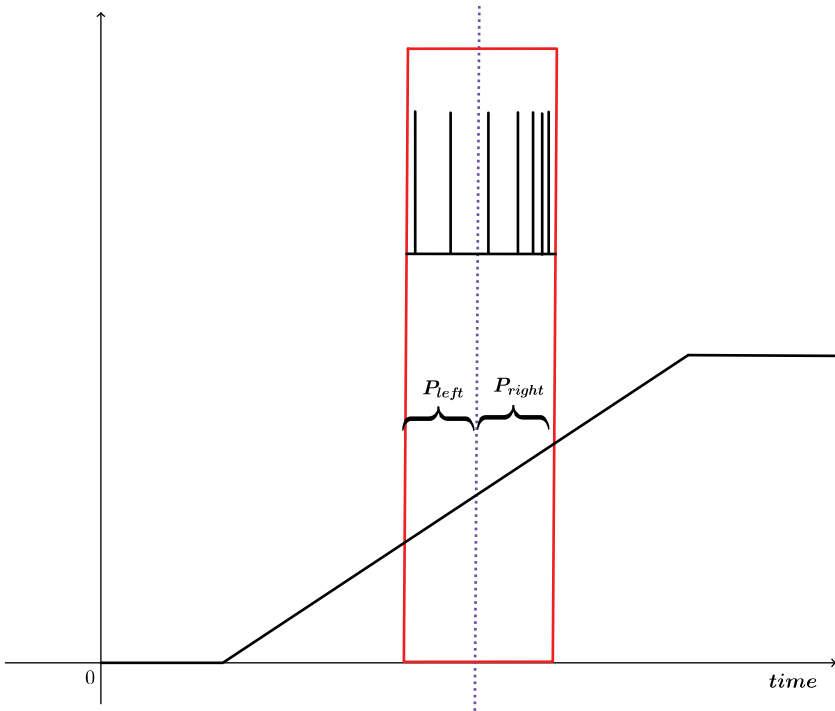


Figure 2: When the postsynaptic rate (y -axis) is rising in time (x -axis), consider a presynaptic spike (blue dotted vertical line) and a window of sensitivity before and after (bold red window). Because the firing probability is greater on the right subwindow, $P_{right} > P_{left}$, one is more likely to find a spike in the right subwindow than in the left subwindow, and it is more likely to be close to the presynaptic spike if that slope is higher, given that when spikes occur on both sides, no weight update occurs. This induces the appropriate correlation between spike timing and the temporal slope of the postsynaptic activity level, which is confirmed by the simulation results of section 4.2.

spikes are randomly drawn from a Poisson process with a rate proportional to $\rho(s_j)$. This assumption ignores spike synchrony effects (which we will do in the rest of this article and which we leave for future work to investigate).

4.1.2 Why It Works. Why does the proposed rate-based weight update mimic the STDP behavior observed during biological experiments? Consider a rising postsynaptic activity level, $\dot{s}_i > 0$, as in Figure 2, and a presynaptic spike occurring somewhere during that rise. Let us assume a Poisson process for the postsynaptic spike as well, with a rate proportional to $\rho(s_i)$. According to this assumption, the rising postsynaptic activity makes

postsynaptic spikes more likely in a fixed time window following the presynaptic spike than in a window of the same length preceding it. Therefore, if only one spike happens in the whole time interval, it is more likely to be after the presynaptic spike, yielding a positive spike timing difference and, thus, a positive weight change. The situation would be symmetrically reversed if the activity level was decreasing (i.e., $\dot{s}_i < 0$), which would make negative weight changes more likely.

Furthermore, the stronger the slope \dot{s}_i is, the greater will this effect be. A larger slope \dot{s}_i also reduces the expected relative spike timing between the presynaptic spike and stochastically occurring postsynaptic spikes, which for a strong slope possibly occur just after or just before the presynaptic spike. When spikes occur on both sides and with about the same time difference, the effects on $W_{i,j}$ cancel each other.

4.2 Simulation Results. To validate the hypothesis that equation 3.18 (or equation 4.1, respectively) yields a relationship between spike timing difference and weight change that is consistent with biological observations shown in Figure 1, we ran the simulations presented in the following sections.

4.2.1 Method. We simulate random variations in the presynaptic neural firing rate $\rho(s_j)$ as well as random variations in the postsynaptic firing rate $\rho(s_i)$ induced by an externally driven voltage. By exploring many configurations of variations and levels at pre- and postsynaptic sides, we hope to cover the possible natural variations. The Poisson process underlying the spike generation (as described in section 4.1) is approximated by performing a discrete-time simulation with a Bernoulli draw of the binary decision spike-versus-no-spike within the time interval of interest. That is, we generate and record pre- and postsynaptic spikes sampled according to a Bernoulli at each discrete t with probability proportional to $\rho(s_i)$ and $\rho(s_j)$, respectively, and record \dot{s}_i as well, in order to implement either equation 3.18 or a classical nearest-neighbor STDP update rule (Markram, Lübke, Frotscher, & Sakmann, 1997; Bi & Poo, 1998; Morrison, Diesmann, & Gerstner, 2008; Van Rossum, Bi, & Turrigiano, 2000).²

For every presynaptic spike, we consider a window of 20 time steps before and after. If we have one or more postsynaptic spikes in both the left and right windows, no postsynaptic spike at all, or a single postsynaptic spike that coincides with the presynaptic spike, the weight is not updated. Otherwise we measure the time difference between the closest postsynaptic spike (the nearest neighbor) and the presynaptic spike and compute the weight change using the current variable values.

²Python scripts for those simulations are available at http://www.iro.umontreal.ca/%E2%88%BCbengioy/src/STDP_simulations.

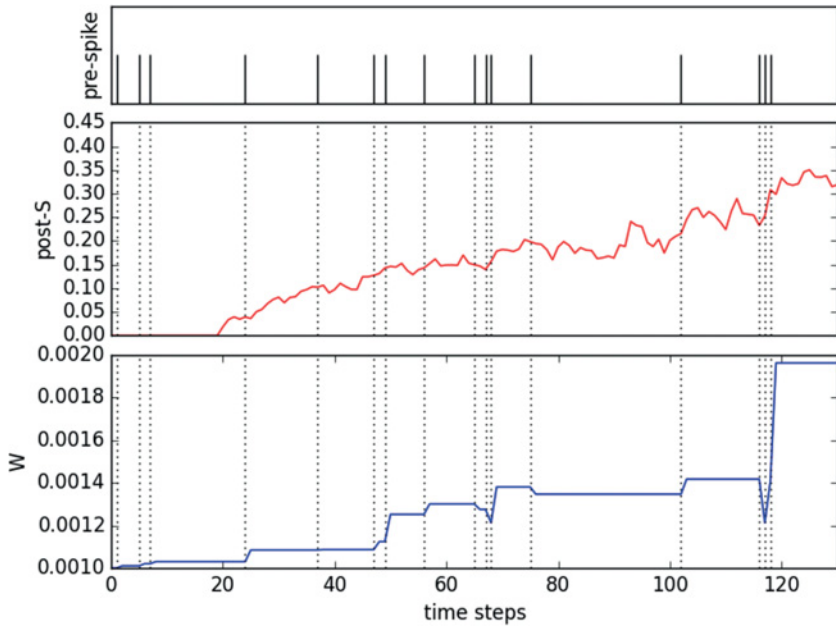


Figure 3: Example of a rate and spike sequence generated in the simulations, along with weight changes according to the spike-dependent variant of our update rule, equation 4.1. (Top) Presynaptic spikes ξ_j (which is when a weight change can occur). (Middle) Integrated postsynaptic activity s_j . (Bottom) Value of the updated weight $W_{i,j}$.

To compute appropriate averages, 500 random sequences of rates are generated, each of length 160 time steps, and 1000 randomly sampled spike trains are generated according to these rates.

For measuring the effect of weight changes, we measure the average squared rate of change $E[||s_t^2||]$ (or, rather, its discrete time approximation $E[|(s_{t+1} - s_t)^2|]$) in two conditions: with weight changes (according to equation 3.18), and without.

4.2.2 Results. An example of spike sequence, the integrated postsynaptic activity, and the corresponding weight values is illustrated in Figure 3.

Figure 4 shows the results of these simulations, comparing the weight change obtained at various spike timing differences for equation 3.18 (left) and for the nearest-neighbor STDP rule (right), both matching the biological data well (see Figure 1).

Figure 5 shows that both update rules are strongly correlated, in the sense that for a given amount of weight change induced by one rule, we observe an average a linearly proportional weight change by the other rule.

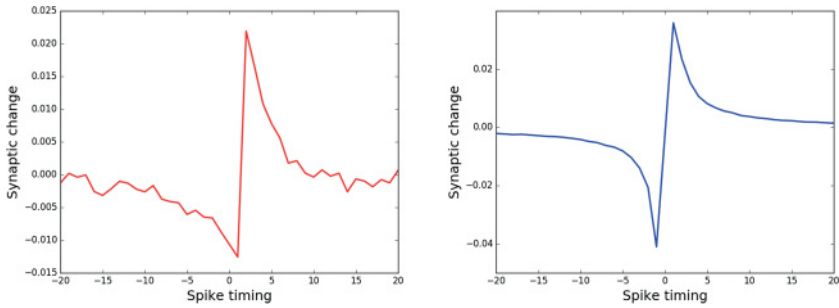


Figure 4: Simulations show that by following the proposed rate-based update rule, we recover the biologically observed relationship between spike timing difference (postsynaptic spike time minus presynaptic spike time, horizontal axis) and the synaptic change (weight update, vertical axis), which is also obtained by the nearest-neighbor rule. (Left) The weight updates are obtained with the proposed update rule, equation 3.18. (Right) The weight updates are obtained using the nearest-neighbor STDP rule. Compare with the biological findings shown in Figure 1.

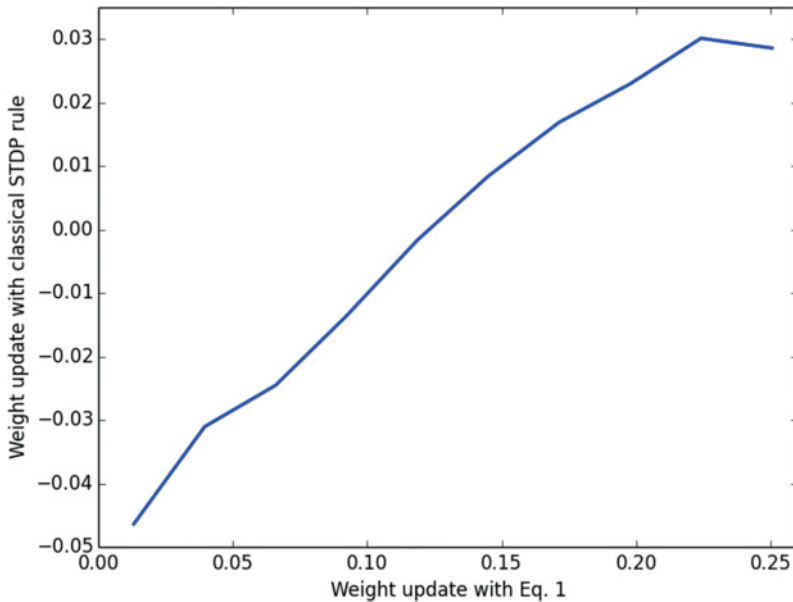


Figure 5: The average STDP update according to the STDP nearest-neighbor rule (vertical axis) versus the weight update according to the proposed rule, equation 3.18 (horizontal axis). We see that on average over samples (by binning values of the x -axis values), the two rules agree closely, in agreement with visual inspection of Figure 1.

5 Related Work

An important inspiration for this work is the idea proposed by Hinton (2007) that brains can implement backpropagation by using temporal derivatives to represent activation gradients and the suggestion that combining this assumption with STDP would approximately yield SGD on the synaptic weights.

The idea of neural computation corresponding to a form of stochastic relaxation toward lower energy configurations is, of course, very old—for example with the Boltzmann machine (Hinton & Sejnowski, 1986) and its Gibbs sampling procedure. (For more recent work in this direction, see also Berkes et al., 2011.) What differs here from the Boltzmann machine is that we consider the state space to be continuous (associated with the expected voltage potential, integrating out the random effects due to spikes) rather than discrete and that we consider very small steps (going down the gradient of the energy), which is more like a Langevin MCMC, rather than allowing each neuron to stochastically jump with higher probability to its optimal state, given its neighbors' configuration, which is what Gibbs sampling does.

The closest work to the STDP-related weight update proposed here is probably that of Xie and Seung (2000), which we have already mentioned. Notable additions to this work brought by our analysis include demonstrating that the relationship of spike timing to weight change is a consequence of equation 3.18, rather than the other way around (and a small difference in the use of $\dot{s} = ds/dt$ instead of $d\rho(s)/dt$).

The proposed rule can be contrasted with theoretical synaptic learning rules, which tend to be based on the Hebbian product of pre- and postsynaptic activity, such as the BCM rule (Bienenstock, Cooper, & Munro, 1982; Intrator & Cooper, 1992). The main difference is that the rule studied here involves the temporal derivative of the postsynaptic activity rather than the actual level of postsynaptic activity.

There are, of course, many other papers on theoretical interpretations of STDP, and there are many references in Markram et al. (2012). But more work is needed to explore the connection of STDP to an unsupervised learning objective that could be used to train not just a single-layer network (like PCA and traditional Hebbian updates) but also a deep unsupervised model. Many approaches (Fiete & Seung, 2006; Rezende & Gerstner, 2014) rely on variants of the REINFORCE algorithm (Williams, 1992) to estimate the gradient of a global objective function (basically by correlating stochastic variations at each neuron with the changes in the global objective). Although this principle is simple, it is not clear that it will scale to very large networks due to the linear growth of the variance of the estimator with the number of neurons, which makes it tempting to explore other avenues.

6 Open Questions and Future Work

This work is only a stepping-stone on the way to a complete theory of learning that is both biologically plausible and makes sense from a machine learning perspective. Many open questions remain to be answered, especially on the side of biological plausibility.

If the energy function proposed in equation 2.7 is closer to a biological truth than the energy defined for continuous Hopfield networks (Hopfield, 1984), we should see that the average soma voltage potential is insensitive to changes in the input firing rates when the neuron is saturated (completely turned off or maximally turned on) and synaptic weight changes should also vanish under this condition, as seen by inspection of equation 3.18. It would clearly be interesting to test these predictions in actual biological experiments.

Another open question is how to reconcile the need for symmetric weights when we introduce an energy function and the fact that $W_{i,j}$ and $W_{j,i}$ are stored at two physically different places in biological neurons. An encouraging observation is that earlier work on autoencoders empirically showed that even when the forward and backward weights are not tied, they tend to converge to symmetric values, and in the linear case the minimization of reconstruction error automatically yields symmetric weights (Vincent, Larochelle, Lajoie, Bengio, & Manzagol, 2010). Another encouraging piece of evidence, also linked to autoencoders, is the theoretical result from Arora, Liang, and Ma (2015), showing that the symmetric solution minimizes the autoencoder reconstruction error between two successive layers of rectifying (ReLU) units. In addition, Lillicrap et al. (2014) show that even when the feedback weights do not match the feedforward weights, backpropagation still works because the feedforward weights end up aligning better with the feedback weights through feedback alignment. This suggests that the strict weight symmetry condition can be relaxed without hurting the learning procedure.

While the work we have presented focuses on supervised learning, much remains to be done to obtain a complete probabilistic theory of unsupervised learning that is consistent with STDP. We believe that we have put interesting ingredients in place and that some of our ideas here could be helpful for building such a theory.

Many open questions also remain in the field of neural plasticity. STDP describes only one view of synaptic plasticity, and it may be different in different neurons and preparations. A particularly interesting type of result to consider in the future is about the relationship between more than two spikes induced statistically by the proposed STDP rule. It would be interesting to compare the statistics of triplets or quadruplets of spike timings and weight change under the proposed rule against the corresponding biological observations (Froemke & Dan, 2002).

Future work should also attempt to reconcile the theory with neuroscience experiments showing more complex relationships between spikes and weight change, involving factors other than timing, such as firing rate, of course, but also synaptic cooperativity and depolarization (Feldman, 2012). In addition to these factors, some synaptic changes do not conform to the STDP pattern, in either sign (anti-Hebbian STDP) or nature (e.g., incorporating a general Hebbian element as in the BCM rule). In this regard, although this work suggests that some spike timing behavior may be explained purely in terms of the rates trajectory, it remains an open question if (and where) precise timing of spikes remains essential to explain network-level learning behavior from a machine learning perspective. It is interesting to note in this regard how the BCM rule was adapted to account for the relative timing of spike triplets (Gjorgjieva, Clopath, Audet, & Pfister, 2011).

In artificial neural networks, communication between neurons usually corresponds to real-valued messages. While this works very well in artificial neurons, it does not fit its biological counterpart. However, connections can be drawn between these real-valued messages and spiking (i.e., binary) messages. A real value α can be sent with binary messages by sending the value one $\alpha\%$ and the value zero $(1 - \alpha)\%$ of the time, so that the message will then have the right expected value. Recent work actually has shown that stochastically quantizing the activations of neural networks still allowed them to be trained (Hubara, Courbariaux, Soudry, El-Yaniv, & Bengio, 2016). As Hinton (2016) explained, sending noisy binary messages drawn from a Poisson process is similar to dropout, which is a powerful regularizer, allowing one to train networks with more parameters than the number of available examples.

7 Conclusion

This work presents a piece of a theory that aims at closing the gap between learning in artificial and biological neural networks. We started by showing that neural updates based on leaky integration correspond to Langevin MCMC steps in an energy-based model with continuous hidden variables. Thus, neural computation can be seen as performing probabilistic inference in the sense of moving toward states that have higher probability under the energy-based probability distribution associated with the model. Starting from a stationary point and letting visible units receive an outside driving force pushing them toward some target value, early steps of neural computation correspond to propagating error gradients into internal layers, similar to backpropagation in supervised artificial neural networks. The propagated error gradients can be turned into a weight update that performs stochastic gradient descent minimizing the difference between the state of the visible units and the target value. Scellier and Bengio (2016) have actually shown through experiments that it is possible to train deep

supervised neural networks using this approach. They have also extended the theoretical framework presented here in order to prove that after the output units have been nudged toward a better value (according to prediction error) and after the dynamics have settled to a very near fixed point, applying the contrastive Hebbian update similar to the one presented here corresponds to gradient descent on the prediction error.

The resulting weight update is proportional to the rate of change of postsynaptic activity times the presynaptic activity. Interestingly, as we have shown through simulations and a qualitative argument, it yields a behavior similar to the STDP observations in the brain that depend on spike timing differences. On top of underlining the biological plausibility of the analyzed learning process, this result provides an interesting machine learning interpretation for STDP, since artificial neural networks are generally cast in terms of firing rates.

Acknowledgments

We thank Benjamin Scellier, Dong-Hyun Lee, Jyri Kivinen, Jorg Bornschein, Roland Memisevic, and Tim Lillicrap for feedback and discussions, as well as NSERC, CIFAR, Samsung, and Canada Research Chairs for funding.

References

- Alain, G., & Bengio, Y. (2013). What regularized auto-encoders learn from the data generating distribution. In *Proceedings of the International Conference on Learning Representations*. arXiv:1211.4246
- Andrieu, C., de Freitas, N., Doucet, A., & Jordan, M. (2003). An introduction to MCMC for machine learning. *Machine Learning*, 50, 5–43.
- Arora, S., Liang, Y., & Ma, T. (2015). *Why are deep nets reversible? A simple theory, with implications for training* (Technical Report). arXiv:1511.05653
- Berkes, P., Orban, G., Lengyel, M., & Fiser, J. (2011). Spontaneous cortical activity reveals hallmarks of an optimal internal model of the environment. *Science*, 331, 83–87.
- Bi, G.-q., & Poo, M.-m. (1998). Synaptic modifications in cultured hippocampal neurons: Dependence on spike timing, synaptic strength, and postsynaptic cell type. *Journal of Neuroscience*, 18(24), 10464–10472.
- Bi, G., & Poo, M. (2001). Synaptic modification by correlated activity: Hebb's postulate revisited. *Annu. Rev. Neurosci.*, 24, 139–166.
- Bienenstock, E. L., Cooper, L. N., & Munro, P. W. (1982). Theory for the development of neuron selectivity: Orientation specificity and binocular interaction in visual cortex. *Journal of Neuroscience*, 2, 32–48.
- Feldman, D. E. (2012). The spike timing dependence of plasticity. *Neuron*, 75(4), 556–571.
- Fiete, I. R., & Seung, H. S. (2006). Gradient learning in spiking neural networks by dynamic perturbations of conductances. *Physical Review Letters*, 97(4), 048104.

- Friston, K. J., & Stephan, K. E. (2007). Free-energy and the brain. *Synthese*, 159, 417–458.
- Froemke, R. C., & Dan, Y. (2002). Spike-timing-dependent synaptic modification induced by natural spike trains. *Nature*, 416(6879), 433–438.
- Gerstner, W., Kempter, R., van Hemmen, J., & Wagner, H. (1996). A neuronal learning rule for sub-millisecond temporal coding. *Nature*, 386, 76–78.
- Gjorgjieva, J., Clopath, C., Audet, J., & Pfister, J.-P. (2011). A triplet spike-timing dependent plasticity model generalizes the Bienenstock-Cooper-Munro rule to higher-order spatiotemporal correlations. *Proceedings of the National Academy of Sciences, USA*, 108(48), 19383–19388.
- Hinton, G. E. (2007). *How to do backpropagation in a brain*. Invited talk at the NIPS'2007 Deep Learning Workshop.
- Hinton, G. E. (2016). *Can the brain do back-propagation?* Invited talk at Stanford University Colloquium on Computer Systems. <https://www.youtube.com/watch?v=VIRCYbGgHts>
- Hinton, G. E., & McClelland, J. L. (1988). Learning representations by recirculation. In D. Z. Anderson (Ed.), *Neural information processing systems* (pp. 358–366). College Park, MD: American Institute of Physics.
- Hinton, G. E., & Sejnowski, T. J. (1986). Learning and relearning in Boltzmann machines. In D. E. Rumelhart & J. L. McClelland (Eds.), *Parallel distributed processing: Explorations in the microstructure of cognition. Vol. 1: Foundations* (pp. 282–317). Cambridge, MA: MIT Press.
- Hopfield, J. J. (1984). Neurons with graded responses have collective computational properties like those of two-state neurons. *Proceedings of the National Academy of Sciences, USA*, 81, 3088–3092.
- Hubara, I., Courbariaux, M., Soudry, D., El-Yaniv, R., & Bengio, Y. (2016). Binarized neural networks. In D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, & R. Garnett (Eds.), *Advances in neural information processing systems*, 29 (pp. 4107–4115). Red Hook, NY: Curran.
- Intrator, N., & Cooper, L. N. (1992). Objective function formulation of the BCM theory of visual cortical plasticity: Statistical connections, stability conditions. *Neural Networks*, 5, 3–17.
- Lillicrap, T. P., Cownden, D., Tweed, D. B., & Akerman, C. J. (2014). *Random feedback weights support learning in deep neural networks*. arXiv:1411.0247
- Markram, H., Gerstner, W., & Sjöström, P. (2012). Spike-timing-dependent plasticity: A comprehensive overview. *Frontiers in Synaptic Plasticity*, 4(2), 8.
- Markram, H., Lübke, J., Frotscher, M., & Sakmann, B. (1997). Regulation of synaptic efficacy by coincidence of postsynaptic aps and EPSPS. *Science*, 275(5297), 213–215.
- Markram, H., & Sakmann, B. (1995). Action potentials propagating back into dendrites triggers changes in efficacy. *Soc. Neurosci. Abs.*, 21, 2007.
- Morrison, A., Diesmann, M., & Gerstner, W. (2008). Phenomenological models of synaptic plasticity based on spike timing. *Biological Cybernetics*, 98(6), 459–478.
- Rezende, D. J., & Gerstner, W. (2014). Stochastic variational learning in recurrent spiking networks. *Frontiers in Computational Neuroscience*, 8(38).
- Scellier, B., & Bengio, Y. (2016). *Equilibrium propagation: Bridging the gap between energy-based models and backpropagation*. arXiv:1602.05179

- Shepherd, G. M. (2003). *The synaptic organization of the brain*. New York: Oxford University Press.
- Van Rossum, M. C., Bi, G. Q., & Turrigiano, G. G. (2000). Stable Hebbian learning from spike timing-dependent plasticity. *Journal of Neuroscience*, *20*(23), 8812–8821.
- Vincent, P. (2011). A connection between score matching and denoising autoencoders. *Neural Computation*, *23*(7), 1661–1674.
- Vincent, P., Larochelle, H., Lajoie, I., Bengio, Y., & Manzagol, P.-A. (2010). Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *J. Machine Learning Res.*, *11*, 3371–3408.
- Williams, R. J. (1992). Simple statistical gradient-following algorithms connectionist reinforcement learning. *Machine Learning*, *8*, 229–256.
- Xie, X., & Seung, H. S. (2000). Spike-based learning rules and stabilization of persistent neural activity. In S. Solla, T. Leen, & K. Müller (Eds.), *Advances in neural information processing systems*, *12* (pp. 199–208). Cambridge, MA: MIT Press.

Received September 19, 2015; accepted November 2, 2016.